

REMARKS

This is in response to the Office Action mailed on September 17, 2008. Claims 1-17 were pending in the Office Action, the Examiner rejected all claims.

On page 6 of the Office Action, the Examiner rejected claims 1, 6, 11, 12 and 16 under 35 U.S.C. §103(a) as being unpatentable over Lisle et al. US Patent No. 4,843,389 in view of Katayama et al. US Patent No. 6,260,051. Of the rejected claims, claims 1, 6 and 11 are independent claims. With this amendment, claims 1, 6, 11-13 and 15-16 are amended, and the remaining claims are unchanged in the application.

The present invention involves the collation of words or symbols. Collation means the sorting of text strings (which consist of symbols) according to an order of the symbols that is culturally correct to users of that language. Collation is used when users order linguistic data or perform a search for ordered linguistic data. For example, in the United States, words are collated such that those beginning with the letter "Q" are ordered after those beginning with the letter "P". In other languages, such as Chinese, linguistic symbols may be sorted by phonetic pronunciation or by the number of strokes in a symbols.

A compression is a special group of symbols that is treated as a single sort element. For example, in the Hungarian language, the letters "DZS" form a single sort element, as do the symbols "DZ". These are both "compressions" as used in the present specification. The compression "DZS" is treated as a single sort element and is arranged, in collation order, before the letter "E" in the Hungarian language, but after the compression "DZ".

The compression type of a given compression refers to the number of symbols that are grouped together as a single sort element. For instance, the compression type of the compression "DZS" is 3 (or 3-to-1). The compression type of the compression "DZ" is 2 (or 2-to-1).

The highest compression type used in a given language varies based on the given language. For instance, some languages, such as Bengali or Tibetan, use compression types as high as 8-to-1. It is therefore very difficult to perform collation because, in order to do so, the

collator must first determine whether some of the symbols in an input string to be collated are part of a compression, and should therefore be grouped together as a single sort element.

The present system deals with building tables that can be used in order to identify or sort compressions. One embodiment sets up symbol tables that include a list of code points that uniquely identify one of the symbols, and a sort weight for each of the identified symbols. The compression tables include a compression type that identifies a number of symbols in a given compression, and compressions of symbols of that compression type. Each compression is a grouping of two or more symbols that is treated as a single sort element for purposes of linguistic sorting such that an order in the linguistic sorting is determined based on the type of a given compression, a first of the two or more symbols in the given compression, and a predefined order of those symbols. Thus, for instance, the symbol tables have compressions that begin with the letter “D” sorted before those that begin with the letter “E”. Each compression that begins with “D” is also sorted based on the compression type (i.e., based on the number of symbols contained in the compression).

This is simply neither taught nor suggested by either of the references cited by the Examiner.

The Lisle reference involves dictionaries. Each dictionary is collated, with special characters first, followed by the alphabet, and then by the numbers. See column 15, lines 45-51.

The dictionaries have two entries per word, in collation order. The first entry is the word length and the second entry is the word itself. The dictionary is also segmented so that the entire dictionary need not be pulled up each time a word is to be located in the dictionary. Instead, each segment has a certain number of words in it. Therefore, there is an index of the segments of the dictionary. By accessing the index, the user need not pull up the entire dictionary into memory, but only the relevant segment identified by the index.

The index has an entry for the first word in the identified segment of the dictionary and an entry for the last word in that segment. For example, a dictionary segment may begin with the word “apple” and end with the word “atrophy”. All the words between “apple”

and “atrophy” (inclusive) are in that segment. Both of these entries (“apple” and “atrophy”) would thus be included in the index to identify that segment of the dictionary.

Lisle discusses two embodiments in which the dictionary may either have two byte or single byte entries. Columns 5 and 6 discuss how many bit patterns can be used for control, and how many for word entries in a given dictionary. However, this is just discussing the size of the dictionary and not the individual entries in the dictionary. At this portion of Lisle, Lisle states that one bit is used to determine whether the dictionary is a one byte or two byte configuration. This has nothing to do with the number of symbols in a compression. Instead, it is a discussion of the number of bytes which can be used to define each entry for a word or number or special character in the dictionary, regardless of how many symbols that word or number of special character is composed of. See column 15, lines 30-37, column 4, lines 48-55, column 5, line 21-column 6, line 50.

Lisle also discusses compression. However, to the extent Lisle discusses compression, Lisle is referring to compressing data in the dictionary, and not treating multi-letter elements (of multi-symbol elements) as a single sort element when performing collation.

Therefore, Lisle does not teach compression tables, or sorting compression tables, at all. Similarly, Lisle does not even teach or suggest a “compression type” which identifies a number of symbols in a given “compression” wherein “each compression is a grouping of two or more symbols treated as a single sort element for purposes of linguistic sorting”.

In contrast, independent claim 1 specifically includes “providing a plurality of compression tables, each compression table...having a compression type identifying a number of symbols in a given compression and containing compressions of symbols of that compression type, each compression being a grouping of two or more symbols treated as a single sort element for purposes of linguistic sorting such that an order of a given compression in the linguistic sorting is based on a compression type of the given compression, a first of the two or more symbols in the given compression, and a predefined order of symbols...”. Claim 1 also

specifically includes “sorting the compression tables based on the sort elements, and identifying a highest compression type beginning with the symbols identified by said each code point.”

Lisle simply fails to teach or suggest the idea of a compression which is a multi-symbol sort element that is sorted before or after other sort elements based on the first symbol and then sorted within those similar compressions based on the compression type. This is simply neither taught nor suggested by Lisle.

Independent claim 6 also teaches a method of building a symbol table that includes providing a plurality of compression tables, each compression table...having a compression type and containing compressions of symbols of that compression type, a compression type identifying a number of symbols in a compression, and each compression being a grouping of two or more symbols treated as a single sort element for purposes of linguistic sorting.” Claim 6 also includes “for each code point in the symbol table, sorting the compression tables to order the compressions and to identify a highest compression type for compressions, the order of the compressions being performed by ordering compressions based on a first of the two or more symbols and then ordering the compressions based on compression type, beginning with the symbol identified by said each code point.” Again, Lisle neither teaches nor suggests a concept of a compression as a multi-symbol sort element nor does it teach or suggest ordering of compressions as set out in independent claim 6. This idea is simply missing from Lisle.

Similarly, independent claim 11 is drawn to carrying out linguistic sorting. Claim 11 includes “receiving an input string containing a plurality of letters used in a given language; for a first letter in a combination of letters in the input string, referencing a symbol table to obtain a highest compression type for compressions beginning with said first letter, each compression being a grouping of two or more letters treated as a single sort element for purposes of linguistic sorting and the compression type identifying a number of letters in a given compression, the symbol table having a list of code points each uniquely identifying a letter and a sort weight for the letter identified by said each code point...” Lisle simply fails to teach or suggest a multi-

symbol element being treated as a single sort element. Instead, the alphabetization set out in Lisle treats each individual letter separately. It does not even discuss the idea of a compression, much less a compression type, or how to perform linguistic sorting based on the compression and compression types. Applicant thus submits that independent claim 11 is allowable over Lisle.

The deficiencies are not cured by Katayama. Katayama deals with information retrieval. Katayama matches a query character string against the letter strings in a document in an information retrieval process. See column 1, lines 7-18. In doing so, Katayama calculates the occurrence frequency of character strings in the documents and specifically refers to general characters, and special characters. General characters are characters such as letters, while special characters have no meaning but divide the text string of the document into two different strings each of which have meaning by themselves. For instance, a special character may be a space that divides two words in the textual string. See column 4, lines 49-52.

In order to determine a relationship between queries and documents, Katayama segments text in a document into character sequences and counts the frequency of occurrence of various character sequences in the document. Column 1, lines 23-53. The Katayama reference is primarily directed to determining how to accurately calculate the occurrence frequencies of various characters and character combinations.

One of the problems encountered by Katayama is that special characters, such as spaces for example, will have a very high number of occurrences in any document, regardless of the words that appear in that document. Therefore, if the occurrence frequency of spaces alone were considered in determining whether the query relates to the document, that will yield false results. See column 4, lines 33-58. Katayama therefore, does not calculate the occurrence frequency of just special characters, such as spaces. Instead, it calculates the occurrence frequency of those characters in combination with general characters so that the count is not skewed in favor of the special characters. Column 5, lines 10-57.

Specifically, in performing matches between the query and the document, Katayama does not look for matches for the special characters, when they occur as fore or aft

characters, but only when they are sandwiched between general characters. The frequency of occurrence for such a three character chain does not include a frequency for the special character, but is only based on the fore and aft general characters in the three character chain. See column 68.

It will also be noted that Katayama unfortunately uses the word “collating” but it would appear that Katayama does not use that word to mean “ordering”. Instead, Katayama appears to be using the word collating to mean “matching”. For instance, beginning at column 4, line 61, “an object of the present invention is to provide...a conventional character string collating apparatus...in which all pieces of character data of a text are recorded...and a character string collating apparatus in which a retrieval character string is efficiently collated with a registration character string of a text...”. Katayama also states, beginning at column 3, lines 46 “an occurrence frequency collating unit 14 for collating the occurrence frequency of the fore character in each occurrence frequency set...with that of the rear character in a particular occurrence frequency set of another particular two character chain type...”. In these places, it would appear that Katayama is discussing how the occurrence frequency of various characters are matched against one another, and not ordered. In the first citation, it appears that Katayama is discussing how the retrieval character string [i.e., the query] is efficiently matched with (“collated with”) a registration character string of a text (i.e., the text of a document). In the second citation, it appears that the occurrence frequency of a fore character in one character chain is matched against the occurrence frequency of the rear character in another character chain. Thus, it does not appear that Katayama is even directed to linguistic ordering as set out in the present claims.

In any case, Katayama is explicitly silent as to the concept of a compression which is a grouping of two or more symbols that are treated as a single sort element for purposes of linguistic sorting. Similarly, Katayama is silent as to even discussing the concept of a compression type which identifies a number of symbols in a given compression. Yet these things are specifically set out in independent claims 1, 6 and 11, as discussed above. Applicant thus

submits that Katayama does not remedy the deficiencies of Lisle and therefore the claims are allowable over the combination of references cited by the Examiner.

In conclusion, Applicant submits that independent claims 1, 6 and 11 are allowable. Applicant further submits that dependent claims 2-5, 7-10, and 12-17, which depend either directly or ultimately from the independent claims, are allowable as well. Reconsideration and allowance of claims 1-17 are respectfully requested.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 23-1123.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: /Joseph R. Kelly/
Joseph R. Kelly, Reg. No. 34,847
900 Second Avenue South, Suite 1400
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312

JRK:slg